

Bakalářská práce

SYSTÉM POČÍTAČOVÉ ALGEBRY SAGE



Poděkování

Chtěl bych poděkovat RNDr. Romanu Plchovi, Ph.D. za vedení bakalářské práce, trpělivost, cenné rady a připomínky při zpracování daného tématu.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval sám, pouze za pomoci RNDr. Romana Plcha, PhD. a uvedené literatury.

V Brně dne 30. května 2009

Martin Haupt

Obsah

Úvod	6
1 Základní informace o systému Sage	7
1.1 Instalace a spuštění pod Windows	7
1.2 Instalace a spuštění pod Linuxem	8
1.3 Nápověda	9
1.4 Možnosti práce se Sagem	9
1.4.1 Terminál	9
1.4.2 Grafické rozhraní	10
1.4.3 Sagenb.org	12
2 Základy práce se Sagem	13
2.1 Jednoduché operace	13
2.2 Práce s proměnnými	14
2.3 Seznamy	15
2.4 Možnosti výstupu	16
2.5 Interakce s ostatními CAS	17
2.6 Balíček Sagetex	17
3 Funkce	18
3.1 Základní funkce	18
3.2 Definice funkcí	19
4 Matematická analýza	20
4.1 Limity	20
4.2 Derivace	20
4.3 Taylorovy polynomy	21
4.4 Integrály	21
4.5 Diferenciální rovnice	22
5 Základy grafiky	23
5.1 Grafy funkcí jedné proměnné	23
5.2 Grafy funkcí dvou proměnných	26
5.3 Interaktivní grafika	28

6	Lineární algebra	29
6.1	Vektory	29
6.2	Matice	29
6.3	Řešení rovnic a jejich soustav	31
	Závěr	32
	Seznam použité literatury	33

Úvod

Systém počítačové algebry (anglicky Computer algebra system, zkráceně CAS) je software určený pro výpočty operující se symboly jako reprezentanty matematických objektů, kterými mohou být jak celá, reálná či komplexní čísla, tak například i polynomy, funkce, matice či soustavy rovnic.

První systém počítačové algebry se objevil již na začátku 60. let 20. století. Průkopníkem na tomto poli byl holandský fyzik a držitel Nobelovy ceny Martinus J. G. Veltman se svým programem Schoonship, představeným již roku 1963. Mezi první rozšířenější systémy počítačové algebry pak patřily programy Reduce (vývoj již v 60. letech 20. století), Derive (1988 až 2007, poslední vyšlá verze Derive 6.1) a Mascyma (1968 až 1982, mnoho inovací bylo použito i v programech z pozdějších let).[1]

Dnes je na trhu řada open source a komerčních programů. V případě komerčních programů jsou na špičce Mathematica a Maple, v oblasti open source řešení patří mezi nejznámější Axiom, Maxima a právě Sage, kterým se zabývá tato práce.

První verze Sage vyšla v únoru 2005 a za jejím vývojem stojí americký matematik William A. Stein¹, v současné době profesor matematiky na University of Washington. O další vývoj se starají především studenti pod jeho vedením. Sage je tvořen v programovacím jazyce Python² a již od počátku je šířen jako open source pod všeobecnou veřejnou licenci GNU³. Cílem autorů je „vytvoření schopné volně šiřitelné open source alternativy pro programy Magma, Maple, Mathematica a Matlab“. V současné době je k dispozici v různé podobě pro většinu platforem (Linux, Microsoft Windows, Solaris, Apple Max OS X) a prochází neustálým vývojem, kdy nové verze s většími či menšími změnami vycházejí přibližně jednou za měsíc (aktuální verze 3.4.2 vyšla 5. května 2009). Všechny potřebné informace (návody, odkazy ke stažení, přídatné balíčky atd.) jsou k nalezení na oficiálních stránkách projektu <http://www.sagemath.org/>.

¹Osobní stránky <http://www.wstein.org/>.

²Mezi další známější programy vytvořené v jazyce Python patří především volně šiřitelný nástroj pro 3D modelování Blender (<http://www.blender.org/>) a diagnostický a monitorovací program pro Windows Everest(<http://www.lavalys.com/>).

³<http://www.gnu.org/licenses/gpl.html>

Kapitola 1

Základní informace o systému Sage

1.1 Instalace a spuštění pod Windows

Vzhledem k linuxové povaze systému Sage je třeba jej pod Windows provozovat za pomoci tzv. virtuálního stroje¹. K tomu dobře poslouží autory doporučený software VMware². Pro naše potřeby bude stačit i jeho nejnížší freeware verze VMware Player, která je k dispozici na adrese <http://www.vmware.com/download/player/>. I pro její stažení je však potřebné vyplnění určitých osobních údajů (jméno, e-mail atd.), proto je vhodnější si instalační soubor pro VMware Player stáhnout odjinud, např. ze serveru Slunecnice.cz³ (přímá adresa <http://www.slunecnice.cz/sw/vmware-player/>).

Samotný Sage se pak stáhne z oficiálních stránek projektu, z adresy http://www.sagemath.org/bin/microsoft_windows/ jako obraz pro práci právě v programu VMware. Po rozbalení archivu (pozor na dostatek místa, archiv sám zabírá kolem 700 MB zatímco velikost rozbaleného obrazu se pohybuje kolem 3 GB v závislosti na verzi) ho načteme do VMware Playeru pomocí příkazu „Open“ a můžeme s ním pracovat.

Po spuštění systému Sage (za spuštění považujeme načtení staženého obrazu pomocí programu VMware) se zobrazí následující:

```
Welcome to Sage: http://www.sagemath.org
```

```
Type one of the following.
```

```
notebook -- start the SAGE notebook server
```

```
off      -- turn this machine off
```

¹„Virtuální stroj je v informatice software, který vytváří virtualizované prostředí mezi platformou počítače a operačním systémem, ve kterém koncový uživatel může provozovat software na abstraktním stroji.“[2]

²<http://www.vmware.com/>

³<http://www.slunecnice.cz/>

```

manage  -- (ADVANCED) type "sudo du" to become "root".
          Type sage at the command line to run the sage
          command line.

sage     -- run SAGE command line

sage login: _

```

Za `sage login:` poté vyplníme naši volbu. Napsání příkazu `notebook` nám umožní pracovat v grafickém rozhraní zobrazeném v internetovém prohlížeči, příkaz `off` ukončí práci s tímto virtuálním strojem, respektive přímo Sagem. Volba `manage` umožní pracovat přímo se systémem, který je obsažen v obrazu Sage pro VMware, jako administrátor. Zároveň umožňuje práci se Sagem jako bychom pracovali přímo v terminálu pod Linuxem. A nakonec `sage` spustí Sage přímo z příkazové řádky našeho virtuálního stroje.

V současné době už probíhají práce na nativní verzi pro Windows (projekt nese prostý název Windows Sage), takže snad v brzké době nebudeme nadále muset pod Windows využívat virtualizaci a práce se Sagem se tedy ještě zjednoduší. Informace o probíhajícím vývoji jsou na adrese <http://wiki.sagemath.org/windows> a <http://windows.sagemath.org/>, kde je v době tvorby této práce již ke stažení rozpracovaná verze.

1.2 Instalace a spuštění pod Linuxem

Na stránce <http://www.sagemath.org/bin/linux/> jsou ke stažení instalační balíčky pro různé linuxové distribuce a to jak v 32bit tak v 64bit verzích. Pro distribuci Ubuntu je Sage ke stažení dokonce pro platformu Intel Atom používané především v tzv. netboocích. Stažený balíček pak jen rozbalíme příkazem `tar zxvf sage-*.tar.gz` (název rozbalovaného souboru se samozřejmě liší podle příslušné verze).

Spuštění provedeme napsáním příkazu `./sage` v adresáři, kde máme Sage nainstalován. Pokud chceme spouštět jednoduše napsáním příkazu `sage` do terminálu, můžeme toho lehce dosáhnout jeho zdefinováním pomocí příkazu `alias` a to např. `alias 'sage'='/home/username/sage-*/sage'` (samozřejmě záleží do jakého adresáře si uživatel Sage nainstaloval).

Po spuštění Sage se objeví následující text obsahující informace o verzi a jejím datu vydání.

```

-----
| Sage Version 3.4.2, Release Date: 2009-05-05           |
| Type notebook() for the GUI, and license() for information. |
-----

sage:

```

Hned po spuštění můžeme se Sagem pracovat přímo z terminálu, v případě, že chceme pracovat v grafickém prostředí, stačí napsat příkaz `notebook()` a Sage nám sám otevře výchozí prohlížeč na příslušné lokální adrese.

Na Ústavu matematiky a statistiky Přírodovědecké fakulty MU je Sage nainstalován na serveru Bart, odkud se dá spustit jednoduše příkazem `sage`.

Pod Linuxem můžeme Sage rozšířit pomocí přídatných balíčků, které přidávají další funkce a můžeme je nalézt na stránce <http://www.sagemath.org/download-packages.html>. Jejich instalace se provádí spuštěním příkazu `sage -i` (např. `sage -i nzmath-0.6.0`). Pokud jsme balíček sami stáhli a uložili do adresáře `sage/local/bin/sage-spkg`, Sage jej ihned nainstaluje, případně se jej sám pokusí z oficiálních stránek stáhnout.

1.3 Nápověda

Na oficiálních stránkách projektu Sage <http://www.sagemath.org/> je k nalezení spousta užitečných materiálů. Pro základní seznámení s programem a jeho funkcemi dobře poslouží tutoriál (<http://www.sagemath.org/doc/tutorial/index.html>) obsahující především dostatek příkladů použití jednotlivých základních funkcí. Pro podrobnější seznámení s možnostmi, které Sage nabízí, pak výborně slouží referenční manuál na adrese <http://www.sagemath.org/doc/reference/>, který stejně jako tutoriál obsahuje množství příkladů, je však mnohem podrobnější a popisuje dobře i různé parametry funkcí. Pod odkazem <http://www.sagemath.org/help.html> najdeme stránku obsahující .pdf soubory s návody, odkazy na různá diskuzní fóra zabývající se Sagem, tzv. screencasty (videa ukazující a popisující práci se systémem Sage) atd.

Vše zmíněné je samozřejmě v anglickém jazyce, v době psaní této práce jsem na internetu nenašel žádné materiály ani diskuzní fóra zabývající se Sagem v češtině či slovenštině.

Přímo v rámci práce s programem je velmi užitečná klávesa Tab, kdy stačí napsat počáteční písmena názvu hledané funkce a po stisknutí Tab Sage buď vypíše seznam všech funkcí takto začínajících a nebo, pokud je jen jediná možnost, název přímo doplní. Další pomoc, kterou Sage při práci nabízí, je výpis informací o jakémkoliv objektu (proměnné, funkci atd.) za použití otazníku. Pokud tedy napíšeme např. `integral?`, Sage nám vypíše informace o typu funkce, balíčku, kterého je součástí, její syntaxi, slovní popis a příklady použití. Pro ještě podrobnější informace pak můžeme použít otazníky dva, tedy např. `integral??`.

1.4 Možnosti práce se Sagem

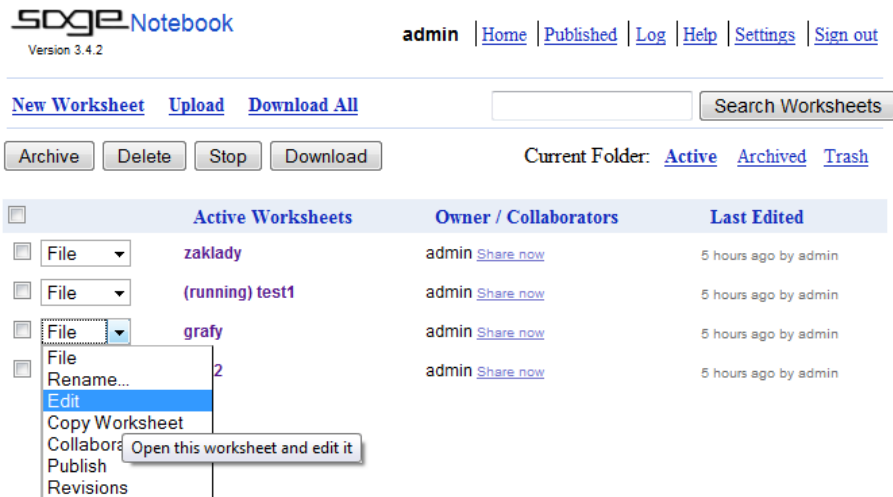
1.4.1 Terminál

Očekávaným způsobem zadáváme příkazy přímo do terminálu a výsledky se zobrazí též na terminál. Vstupní příkazy jsou vždy uvozeny `sage:`, výstup už ne. Příklady příkazů v této práci se tohoto budou pro přehlednost držet. Pokud chceme najednou zadat více příkazů, oddělujeme je středníkem.

Práce se ukončuje příkazem `quit` či `exit`.

1.4.2 Grafické rozhraní

Jak je již zmíněno výše, pro práci v grafickém rozhraní přes internetový prohlížeč je nutno vybrat možnost `notebook`, případně při práci pod Linuxem v terminálu zadat příkaz `notebook()`⁴. Pod Linuxem Sage sám otevře prohlížeč na příslušné stránce, pod Windows nám vypíše adresu, kterou musíme do prohlížeče⁵ sami zadat. Adresa je samozřejmě lokální, takže pro samotnou práci není nutné být připojen k internetu.



Obrázek 1.1: Úvodní obrazovka grafického rozhraní.

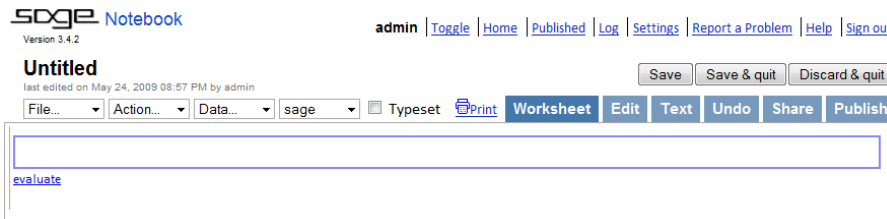
Pro založení nového zápisníku slouží odkaz *New Worksheet*. V nové záložce (případně okně) se nám pak otevře nový prázdný zápisník. Příkazy zadáváme do tzv. buněk (fialově ohraničený obdélník), můžeme jich zadávat více na řádek oddělené středníkem, případně pro přehlednost každý v rámci buňky na vlastní řádek. Odřádkování provádíme klávesou `Enter`, spuštění příkazů provedeme buď stiskem odkazu *evaluate* pod buňkou nebo stiskem kombinace kláves `Shift + Enter` nebo `Ctrl + Enter`. Rozdíl mezi použitím `Shift + Enter` a `Ctrl + Enter` se projeví v případě, že máme v následující buňce již zadány nějaké další příkazy. Pokud v tomto případě použijeme kombinaci s klávesou `Shift`, kurzor skočí do následující buňky, pokud kombinaci s klávesou `Ctrl`, vznikne nám mezi těmito dvěma buňkami nová prázdná buňka. Výsledky se zobrazují do místa mezi jednotlivými buňkami.

V horní části zápisníku vidíme jeho název (pro přejmenování stačí na stávající název kliknout), tlačítko *Save* pro uložení zápisníku, *Save & quit* pro ukončení práce se zápisníkem s uložením změn v něm provedených a *Discard & quit* pro ukončení práce bez uložení provedených změn. Dále jsou zde čtyři vysouvací nabídky: *File...*, *Action...*, *Data...* a *sage*.

⁴Přestože i pod Windows při práci se Sagem v terminálu je možné zadat příkaz `notebook()`, nefunguje zde (alespoň ve verzi 3.4.2) korektně a proto doporučuji raději práci v terminálu ukončit a použít příkaz `notebook`.

⁵Autory je doporučovaný prohlížeč Mozilla Firefox, testoval jsem bez výskytu jakéhokoliv problému i v různých verzích prohlížečů Opera, Google Chrome, Safari i Internet Explorer.

V nabídce *File...* nalezneme možnosti pro načtení zápisníku z uloženého .sws souboru (*Upload worksheet from a file*), vytvoření nového zápisníku (*New worksheet*), tato možnost se spolu s načtením zápisníku z .sws souboru na disku skrývá i pod možností *Upload or create file...* v nabídce *Data...*), uložení zápisníku do .sws souboru na disk (*Download to a file*), vytisknutí zápisníku (*Print*), přejmenování zápisníku (*Rename worksheet*), vytvoření kopie zápisníku (*Copy worksheet*) a smazání právě používaného zápisníku (*Delete worksheet*).



Obrázek 1.2: Nově vytvořený prázdný zápisník

Nabídka *Action...* obsahuje především příkazy ovlivňující samotnou práci se zápisníkem. Příkaz *Interrupt* slouží pro přerušování probíhajícího výpočtu, *Restart worksheet* pak vymaže „paměť“ v rámci zápisníku a *Save and quit worksheet* má stejnou funkci jako tlačítko *Save & quit*, tedy zavře zápisník a uloží změny v něm provedené. Pro vyhodnocení všech buněk slouží *Evaluate all*, *Hide All Output* a *Show All Output* slouží ke skrytí, respektive odkrytí, všech výstupů z buněk, *Delete All Output* všechny tyto výstupy rovnou smaže. Nakonec *One Cell Mode* a *Multi Cell Mode* slouží k přepínání mezi zobrazením jen s jednou buňkou a se všemi.

V nabídce *sage* se pak nachází různé možnosti pro vyhodnocení příkazů v buňkách s různými parametry.

Na pravé straně od popsaných nabídek se pak nachází „přepínače“ pro zobrazení zápisníku, přičemž výchozí zobrazení je *Worksheet*. *Edit* umožní editovat zdrojový kód zápisníku a *Text* přepne zápisník do textové podoby připomínající svým rozložením práci se Sagem v terminálu. Vedle těchto možností je pak tlačítko *Undo* umožňující přístup k různým starším verzím používaného zápisníku. *Share* umožní editovat zápisník i jiným námi vybraným uživatelům a *Publish* pak zviditelní tento zápisník pro všechny uživatele ovšem bez možnosti jej editovat.

Množství výše popsaných funkcí je přístupná také z úvodní domovské stránky, na které jsme si přes odkaz *New worksheet* vytvořili nový zápisník. Na tuto stránku se dostaneme ze zápisníku vždy po jeho uzavření nebo rovnou přes *Home* na horním okraji stránky. Můžeme zde přes odkaz *Upload* načítat zápisníky z .sws souborů na disku, pomocí *Download all* je všechny najednou v .zip archivu na disk uložit, pro přehlednost si označené zápisníky přesunout z aktivních do složky *Archived*, hromadně zápisníky mazat atp. Všechny funkce jsou velmi intuitivní a dopodrobna už je tedy popisovat dále nebudu.

Pro ukončení práce v grafickém prostředí stačí zavřít prohlížeč a v terminálu stisknout kombinaci kláves `Ctrl + C`. Předtím je samozřejmě třeba si rozdělanou práci uložit.

The screenshot shows the Sage Notebook interface. At the top, it says "SAGE Notebook Version 3.4.2" and "admin | Toggle | Home | Published | Log | Settings | Report a Problem | Help | Sign out". Below that, there's a "test1" section with "last edited on May 24, 2009 12:23 PM by admin" and buttons for "Save", "Save & quit", and "Discard & quit". A menu bar includes "File...", "Action...", "Data...", "sage", "Typeset", "Print", "Worksheet", "Edit", "Text", "Undo", "Share", and "Publish". The main content area has a code editor with the expression $f = \sqrt{\cos(x) + \sin(x^3)} - 56 \cdot \exp(2x^3 + x^2) / (\sqrt{2} + 8/9)$. Below the code editor, there are sections for "f", "show(f)", "latex(f)", and "print(f)". The "show(f)" section displays the rendered expression: $\sqrt{\sin(x^3) + \cos(x)} - \frac{56e^{2x^3 + x^2}}{\sqrt{2} + \frac{8}{9}}$. The "print(f)" section shows the LaTeX code: $\sqrt{\sin(x^3) + \cos(x)} - \frac{56e^{2x^3 + x^2}}{\sqrt{2} + \frac{8}{9}}$. The "evaluate" section shows the final rendered expression with superscripts and subscripts.

Obrázek 1.3: Otevřený zápisník.

1.4.3 Sagenb.org

Sage nám dále nabízí ještě jednu užitečnou možnost: práci na kterémkoliv PC s připojením k internetu bez nutnosti instalace potřebného softwaru. Na stránce <http://www.sagenb.org/> si každý (samozřejmě zdarma) může vytvořit vlastní účet a pracovat se zápisníky odkudkoliv přes internetový prohlížeč. Zápisníky se v tomto případě neukládají na lokální disk, ale na server, ze kterého je samozřejmě možno je stáhnout a pracovat s nimi doma či naopak své vlastní zápisníky vytvořené offline uploadovat na server, kde s nimi nejen můžeme pracovat odkudkoliv třeba během pracovních cest, ale také je můžeme sdílet s ostatními uživateli.

Na stránkách <http://www.sagenb.org/pub/> se pak nachází seznam všech zveřejněných zápisníků. Ty jsou přístupné všem uživatelům bez nutnosti registrace.

Kapitola 2

Základy práce se Sagem

2.1 Jednoduché operace

Pro základní operace jako je sčítání, odčítání, násobení atp. se v Sagi používají očekávaná znaménka $+$, $-$, $*$ a $/$. Pro celočíselné dělení pak $//$ a pro zbytek po celočíselném dělení se používá $%$ ¹. Desetinná čísla využívají pro svůj zápis desetinné tečky.

```
sage: 3.5 + 2.4
5.900000000000000
```

```
sage: 7/2*12/5
42/5
```

```
sage: 3.5 - 12/5
1.100000000000000
```

```
sage: 20//7
2
```

```
sage: 20%7
6
```

Umocňování se provádí příkazem `**` nebo `^`. Pro odmocnění stačí dosadit vhodný zlomek, pro druhou odmocninu lze očekávatelně použít příkaz `sqrt()`.

```
sage: 4.5^(sqrt(65))
184657.797180659
```

Pro porovnání či zjišťování rovnosti fungují příkazy `==`, `<=`, `>=`, `<`, `>` a `<>`.

```
sage: g = 25
sage: g
```

¹Je tedy jasné, že pro uvozování komentářů v buňce zápisníku nebude sloužit `%`, ale nějaký jiný symbol. Sage pro uvození komentářů využívá `#`.

25

```
sage: g == 14
False
```

```
sage: sqrt(g) <= 5
True
```

Nakonec pro odkazování se na předchozí výsledky výpočtů slouží podtržítka. V grafickém rozhraní funguje jen odkazování se na výsledek předchozí, v terminálu se můžeme pomoci dvou a tří podtržitek odkázat na výsledek ještě o jednu či dvě úrovně předcházející.

```
sage: h = 587.2
sage: h
587.2000000000000
```

```
sage: h/56
10.4857142857143
```

```
sage: _*2
20.9714285714286
```

```
sage: __*3
31.4571428571429
```

2.2 Práce s proměnnými

Pokud chceme s objektem pracovat jako s proměnnou, je třeba jej takto zadefinovat pomocí funkce `var()`, takže např. `y = var('y')` zadefinuje `y` jako proměnnou a je možno s ní tak dále pracovat. Pokud chceme zadefinovat proměnných více najednou, lze jednoduše `y,z = var('y,z')`. Objekt `x` je jako proměnná zadefinovaný již ve výchozím stavu. Pro přiřazování hodnot proměnným se používá `=`.

Pokud chceme do objektu pouze uložit nějakou konstantní hodnotu (ať už číselnou nebo třeba graf), není nutné ji zadefinovat pomocí funkce `var()`.

```
sage: solve(z^2 + 4*z - 2, z)
Traceback (most recent call last):
...
NameError: name 'z' is not defined
```

```
sage: z = var('z')
sage: solve(z^2 + 4*z - 2, z)
[z == -sqrt(6) - 2, z == sqrt(6) - 2]
```

Co se vyhodnocování proměnných týče, tak Sage vyhodnocuje do hloubky podobně jako např. Maple.

```
sage: a = 2; b = a; c = b; d = c; e = d; f = e
sage: g = 3
sage: h = g - f
sage: h
1
```

2.3 Seznamy

Seznamy patří mezi velmi využívané datové struktury v Sagi, setkáme se s nimi např. při práci s vektory a maticemi, ale i při jednoduchém přiřazování hodnot parametrům funkcí. Zadáváme je do hranatých závorek, jednotlivé prvky seznamu oddělujeme čárkou. Pokud chceme seznam např. celých čísel v určitém rozsahu, zápis můžeme zjednodušit zápisem dvou teček mezi hraniční prvky tohoto seznamu.

```
sage: s = [1.18, 2.35, 56.7, 0.73, 0.52]
sage: s
[1.1800000000000000, 2.3500000000000000, 56.700000000000000,
0.7300000000000000, 0.5200000000000000]
```

```
sage: [-1..5]
[-1, 0, 1, 2, 3, 4, 5]
```

Při zadávání seznamu se objekty vyhodnocují, resp. zjednodušují.

```
sage: t = 5
sage: p = [t, 6, 3 + 4]
sage: p
[5, 6, 7]
```

Na prvky v uloženém seznamu se pak odkazujeme přidáním čísla prvku v hranatých závorkách za název seznamu². Maximální prvek získáme funkcí `max()`, minimální pak `min()`. Počet prvků seznamu nám vrátí funkce `len()`.

```
sage: s[1]
2.3500000000000000
sage: min(s); max(s)
0.5200000000000000
56.700000000000000
sage: len(p)
3
```

²Prvky se v seznamu počítají od „nultého“, pokud tedy chceme získat první prvek seznamu `p`, je třeba napsat `p[0]`.

2.4 Možnosti výstupu

Sage nám umožňuje výsledky na výstup zobrazit různými způsoby za pomoci speciálních funkcí. Běžně je výstup zobrazen v podobě zdrojového kódu Sage, pro lepší přehlednost ale můžeme využít funkcí `show()` a `print()`. Funkce `print()` nám zobrazí výsledek názorněji za pomoci ASCII znaků, je tedy použitelná i v rámci práce v terminálu. `show()` používá k zobrazení balíček `jsMath`³ umožňující zobrazení matematických symbolů v rámci HTML stránek a dávající výsledek velmi podobný textům sázeným v `LATEX`u. Tím se dostáváme k další funkci nazvané prozaicky `latex()` která nám zobrazený objekt převede na zdrojový kód právě pro sazbu systémem `LATEX`.

```
sage: f = sqrt(cos(x)+sin(x**3))-56*exp(2*x**3+x**2)/(sqrt(2)+8/9)
sage: f
sqrt(sin(x^3) + cos(x)) - 56*e^(2*x^3 + x^2)/(sqrt(2) + 8/9)
```

```
sage: show(f)
```

$$\sqrt{\sin(x^3) + \cos(x)} - \frac{56e^{2x^3+x^2}}{\sqrt{2} + \frac{8}{9}}$$

```
sage: latex(f)
```

```
\sqrt{ \sin \left( {x}^{\{3\}} \right) + \cos \left( x \right) } -
\frac{{56 {e}^{\{2 {x}^{\{3\}} \} + {x}^{\{2\}} \} }}{\sqrt{ 2 } +
\frac{8}{9}}
```

```
sage: print(f)
```

$$\sqrt{\sin(x^3) + \cos(x)} - \frac{56 e^{2x^3 + x^2}}{\sqrt{2} + \frac{8}{9}}$$

Pokud chceme výraz zobrazit numericky (např. odmocniny atp.), můžeme na to využít funkci `numerical_approx()`, případně výrazně kratší zápis stejné funkce `n()`. Přesnost numerické aproximace určujeme pomocí dvou volitelných parametrů: `prec` určující na kolik bitů se má výraz spočítat (implicitně 53) a případně `digits` určující počet míst k vyčíslení (pozor, ne desetinných, započítávají se všechna místa).

```
sage: numerical_approx(pi**2, digits = 5)
9.8696
```

```
sage: n(sqrt(pi)*e, 80)
4.8180290946987220571215
```

³<http://www.math.union.edu/~dpvc/jsMath/>

2.5 Interakce s ostatními CAS

Sage nám umožňuje v rámci jeho prostředí využívat funkce ostatních systémů počítačové algebry (zkráceně CAS), které máme v systému nainstalované. Pro u nás na katedře velmi používaný Maple můžeme příkazem `maple('command')` vyvolat námi požadovaný příkaz z Maplu a s výsledkem dále pracovat v Sagi.

Tato možnost se samozřejmě neomezuje jen na Maple, ale lze ji obdobně využít i pro ostatní systémy jako je Axiom, Magma, Matlab, Maxima, Mathematica a další. Na univerzitním serveru Bart toto funguje bez problému např. pro systémy Maxima, Maple a Matlab.

```
sage: maple('factor(x^3 - 1)')
(x-1)*(x^2+x+1)
```

```
sage: maxima('taylor(asin(x), x, 0, 10)')
x+x^3/6+3*x^5/40+5*x^7/112+35*x^9/1152
```

```
sage: print(_)
```

$$\begin{array}{cccccccc} & & 3 & & 5 & & 7 & & 9 \\ & & x & & 3 x & & 5 x & & 35 x \\ /T/ & x + & \frac{\quad}{6} & + & \frac{\quad}{40} & + & \frac{\quad}{112} & + & \frac{\quad}{1152} & + \dots \end{array}$$

Podrobnější popis možností interakce s jinými systémy počítačové algebry je k nalezení v referenčním manuálu, přesněji přímo na stránce <http://www.sagemath.org/doc/reference/interfaces.html>.

2.6 Balíček Sagetex

Pro sázecí systém \LaTeX je k dispozici balíček nazvaný Sagetex⁴ umožňující využívat možností Sage přímo v dokumentu sázeném právě v \LaTeX u. Základní myšlenkou je zjednodušit uživateli tvorbu grafů, rozsáhlejší výpočty atp. při sazbě dokumentu. Pro využití těchto možností je samozřejmě třeba mít stažený tento balíček a zároveň nainstalovaný Sage. Vzhledem k nemožnosti Sage přímo nainstalovat na stanicích s operačním systémem Windows není tedy pro uživatele tohoto systému využití balíčku možné.

Pokud má uživatel nainstalovaný jak Sage, tak připravený balíček Sagetex, může do svých dokumentech psaných v \LaTeX u do příkazu `\sage{}` zadat příkaz ze Sage, např. nějaký výpočet, a při překladu se tento výpočet provede a do výsledného dokumentu se dosadí výsledek. V případě, že chce uživatel vložit např. graf funkce či jinou grafiku vytvořenou Sagem, je třeba použít příkaz `\sageplot{}`. Samotný překlad takového dokumentu se pak skládá ze tří kroků. Nejdříve je třeba dokument přeložit \LaTeX em a při tom vznikne soubor s koncovkou `.sage`. Na ten je pak třeba Sage zavolat a soubor zpracovat (např. `sage pokus.sage`). Poté už stačí naposledy přeložit \LaTeX em a je hotovo.

⁴<http://www.ctan.org/tex-archive/help/Catalogue/entries/sagetex.html>

Kapitola 3

Funkce

3.1 Základní funkce

Pro výpočty hodnot goniometrických funkcí slouží příkazy `sin()`, `cos()`, `tan()` a `cot()`¹, pro cyklometrické pak `arcsin()`, `arccos()`, `arctan()` a `arccot()`.

```
sage: sin(pi/6)
1/2
```

```
sage: arcsin(1/2)
pi/6
```

```
sage: arctan(2.458)
1.18441178085696
```

Exponenciální funkci se základem e zapisujeme jako `exp(x)`.

```
sage: exp(x**2)
e^x^2
```

```
sage: print(exp(x**2))
      2
      x
      e
```

Výpočet logaritmu provedeme příkazem `log()`. Parametry nastavujeme dva, prvním je objekt, na který chceme funkci aplikovat, druhým pak základ logaritmu. Pro přirozený logaritmus lze pak použít funkci `ln()` mající jediný parametr.

```
sage: ln(e^4)
4
```

¹ π se zapisuje jako `pi`, stejně tak pro ostatní písmena řecké abecedy slouží jejich anglické názvy.

```
sage: log(4454, 10)
log(4454)/log(10)
```

```
sage: log(4454.0, 10)
3.64875021269802
```

Pro práci s polynomy obsahuje Sage např. funkci `expand()` pro převedení polynomu na rozšířený tvar. Sage sám polynom zobrazuje v co nejjednodušším tvaru. Pro rozklad na kořenové činitele pak slouží příkaz `factor()`.

```
sage: y = var('y')
sage: p = x^4*y^2 + x^5*y + (x^2 + x*y^3)*(x - y^2) - x^3
sage: p
(x - y^2)*(x*y^3 + x^2) + x^4*y^2 + x^5*y - x^3
```

```
sage: expand(p)
-x*y^5 + x^2*y^3 + x^4*y^2 - x^2*y^2 + x^5*y
```

```
sage: factor(p)
-x*y*(y^4 - x*y^2 - x^3*y + x*y - x^4)
```

```
sage: factor(x^6 - 6*x^3 - 16)
(x - 2)*(x^2 + 2*x + 4)*(x^3 + 2)
```

3.2 Definice funkcí

Vlastní funkce definujeme za použití příkazu `def`. Jeho základní syntaxe je:

```
sage: def jmeno_fce(promenna):
...     kostra funkce
```

V definici můžeme samozřejmě používat základní programátorské funkce jako je např. větvení `if` a `for` cykly.

```
sage: def podil(a,b):
...     if b==0: return 'Pozor, delite nulou...'
...     return a/b
sage: podil(5.0,2)
2.5000000000000000
```

```
sage: podil(5,2)
5/2
```

```
sage: podil(5,0)
'Pozor, delite nulou...'
```

Kapitola 4

Matematická analýza

4.1 Limity

Výpočet limit se provádí příkazem `limit()`, případně `lim()`. Základní syntaxe je `lim(f, x = a)` rovnající se výpočtu limity $\lim_{x \rightarrow a} f(x)$ ¹. Pro výpočet jednostranných limit slouží parametr `dir`, kdy jej pro výpočet limity zprava nastavíme jako `'plus'`, respektive `'above'`, pro výpočet limity zleva pak `'minus'`, respektive `'below'`.

```
sage: b = var('b')
sage: lim(b*sin(x)/x, x = 0)
b
```

```
sage: lim(tan(x)/2, x = pi/2)
und
```

```
sage: lim(tan(x)/2, x = pi/2, dir = 'plus')
-Infinity
```

```
sage: lim(tan(x)/2, x = pi/2, dir = 'minus')
+Infinity
```

4.2 Derivace

Funkce derivujeme pomocí `diff()`, respektive `derivative()`. Pro derivaci funkce `f` jedné proměnné vypadá zápis takto: `diff(f)`. Kvůli přehlednosti je však vhodnější specifikovat i podle které proměnné derivujeme.

```
sage: diff(cos(x^4), x)
-4*x^3*sin(x^4)
```

Výpočet derivací vyšších řádů provádíme pomocí volitelného třetího parametru:

¹Nekonečno se zapisuje pomocí příkazu `infinity` případně pomocí dvou malých „o“, tedy `oo`.

```
sage: diff(cos(x^4), x, 4)
1152*x^8*sin(x^4) - 24*sin(x^4) + 256*x^12*cos(x^4) -
816*x^4*cos(x^4)
```

Parciální derivace mají syntaxi stejnou, je však nutné určit, podle které proměnné derivujeme. Pokud chceme derivace smíšené, opět stačí přidat další parametr.

```
sage: y = var('y')
sage: f = cos(x^y); f
cos(x^y)
```

```
sage: diff(f)
Traceback (most recent call last):
...
ValueError: must supply an explicit variable for an expression
containing more than one variable
```

```
sage: diff(f, x)
-x^(y - 1)*sin(x^y)*y
```

```
sage: diff(f, x, 2)
-x^(2*y - 2)*cos(x^y)*y^2 - x^(y - 2)*sin(x^y)*(y - 1)*y
```

```
sage: diff(f, x, 2, y)
x^(3*y - 2)*log(x)*sin(x^y)*y^2 - 2*x^(2*y - 2)*log(x)*cos(x^y)*
y^2 - x^(y - 2)*log(x)*sin(x^y)*(y - 1)*y - x^(2*y - 2)*log(x)*
cos(x^y)*(y - 1)*y - x^(y - 2)*sin(x^y)*y - 2*x^(2*y - 2)*
cos(x^y)*y - x^(y - 2)*sin(x^y)*(y - 1)
```

4.3 Taylorovy polynomy

Pro výpočet Taylorova polynomu slouží příkaz `taylor()`, zápis `taylor(f, x, a, n)` značí výpočet Taylorova polynomu funkce f proměnné x v bodě a stupně n .

```
sage: taylor(2*sin(x)/x, x, 0, 4)
2 - x^2/3 + x^4/60
```

4.4 Integrály

Integrujeme funkcí `integral()`, respektive `integrate()`. Podobně jako u derivací stačí pro základní výpočet integrálu jako parametr integrovaná funkce, je ovšem lepší si zvyknout psát i proměnnou přes kterou integrujeme.

```
sage: integral(4*x*sin(x^2), x)
-2*cos(x^2)
```

```
sage: y = var('y')
sage: g = y*x^2 + x**5*y*5 + x*log(x)^2
sage: g
5*x^5*y + x^2*y + x*log(x)^2

sage: integral(g, x)
5*x^6*y/6 + x^3*y/3 + x^2*(2*log(x)^2 - 2*log(x) + 1)/4

sage: integral(g, y)
5*x^5*y^2/2 + x^2*y^2/2 + x*log(x)^2*y
```

Určitý integrál se počítá obdobně, navíc přibudou dva argumenty s mezemi.

```
sage: h = 4*x*sin(x^2)
sage: integral(h, x, 0, 5)
4*(1/2 - cos(25)/2)
```

Pokud chceme určitý integrál vyčíslit numericky, můžeme využít funkci `integral_numerical`, která nám na výstup vrátí výsledek a druhou hodnotu udávající maximální chybu.

```
sage: integral_numerical(h, 0, 5)
(0.017594376273053096, 3.5467137744586723e-13)
```

4.5 Diferenciální rovnice

K řešení diferenciálních rovnic slouží funkce `desolve()`. Před samotným řešením diferenciální rovnice je však třeba zadefinovat, že s y budeme pracovat tentokrát jako s funkcí proměnné x . K tomu slouží příkaz `function()` a výsledný zápis pak vypadá takto: $y = \text{function}('y', x)$. Poté již můžeme přistoupit k samotnému řešení rovnice příkazem `desolve()`. Jako první parametr zadáme řešenou rovnici, jako druhý pak řešené proměnné. Celý postup pak bude vypadat takto:

```
sage: y = function('y', x)
sage: desolve(diff(y, x) + y - 1, [y, x])
e^(-x)*(e^x + c)
```

V případě, že máme diferenciální rovnici s počáteční podmínkou, použijeme ještě třetí parametr `ics` do kterého vložíme dvojici počátečních hodnot.

```
sage: y = function('y', x)
sage: desolve(diff(y, x) - cos(x) + y, [y, x], ics = [0, 1])
e^(-x)*(e^x*sin(x) + e^x*cos(x) + 1)/2
```

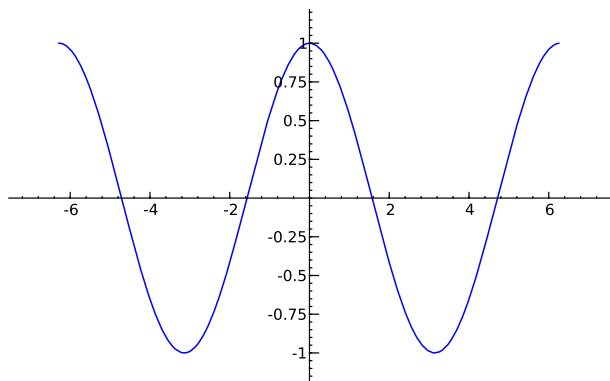
Kapitola 5

Základy grafiky

5.1 Grafy funkcí jedné proměnné

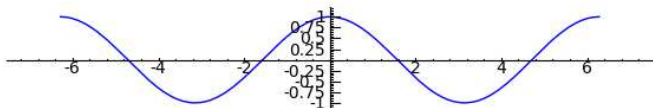
Grafy funkcí se tvoří za pomoci příkazu `plot()`. Jako parametry musíme samozřejmě zadat funkci, jejíž graf chceme vytvořit, a je velmi vhodné i určit následujícími dvěma parametry meze na ose x , v kterých se bude graf pohybovat. Jednoduchý graf funkce pak může vypadat třeba takto:

```
sage: g1 = plot(cos(x), -2*pi, 2*pi); g1
```



Již z prvního grafu je vidět, že Sage sám upravuje poměr os tak, aby se obrázek ideálně vešel. Pokud chceme jednotky na ose x a y ve stejném poměru, lze toho docílit jednoduše pomocí příkazu `show` a nastavení volitelného parametru `aspect_ratio` na hodnotu 1.

```
sage: show(g1, aspect_ratio = 1)
```



Příkaz `show()` nabízí i další užitečné parametry, např. zobrazení grafu bez os dosáhneme nastavením `axes = False`, zobrazení grafu s rámečkem ve stylu Matlabu pak parametrem `frame = True` a nastavení vlastního rozměru grafu

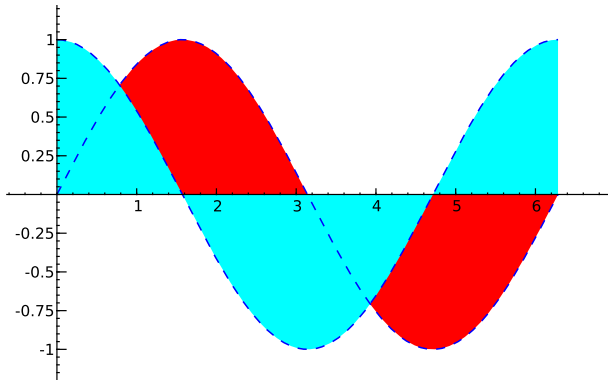
na nastavením parametru `figsize` (např. `figsize = [5,6]`).

Další vlastnosti zobrazení grafů lze nastavit přímo pomocí volitelných parametrů funkce `plot()`. Zde je jejich výběr i s krátkým popisem:

- **color**: Nastavuje barvu křivky. Hodnotou parametru může být trojice hodnot mezi 0 a 1 udávající odstín v RGB kódu (např. `(0,0.5,0.7)`), HTML kód barvy (např. `'#790C11'`) nebo textový řetězec pro anglický název barvy (např. `'yellow'`, samozřejmě možné jen pro konečný počet barev). Implicitní barva je `'blue'`.
- **thickness**: Nastavuje tloušťku linky grafu.
- **alpha**: Nastavuje průhlednost křivky hodnotou mezi 0 a 1.
- **hue**: Umožní nastavit barvu pomocí jediné hodnoty značící odstín.
- **linestyle**: Nastavuje styl linky grafu, pro přerušovanou nastavíme na `'--'`, pro čerchovanou `'-.'` a tečkovanou `':'`. Pokud z nějakého důvodu linku grafu nechceme, hodnotu parametru dáme `''`. Ve výchozím nastavení je linka plně, tzn. hodnota parametru je `'-'`.
- **fill**: Hodnota `"axis"`, případně `True`, vyplní plochu mezi grafem a osou `x`, `"min"` a `"max"` plochu pod, respektive nad, grafem. Pokud zadáme jako parametr jinou funkci, vyplní se plocha mezi linkou zobrazované funkce a funkce z parametru.
- **fillcolor**: Umožňuje nastavit barvu výplně podobně jako parametr `color` nastavuje barvu křivky. Ve výchozím stavu je nastaveno na `automatic`, kdy se barva přizpůsobuje počtu funkcí v grafu atp.
- **fillalpha**: Nastavuje průhlednost výplně hodnotou mezi 0 a 1. Výchozí nastavení je 0.5.
- **zorder**: V případě zobrazení více grafů do jednoho obrázku určuje jeho pozici na ose `z`. Čím nižší číslo, tím více „vepředu“ graf je.

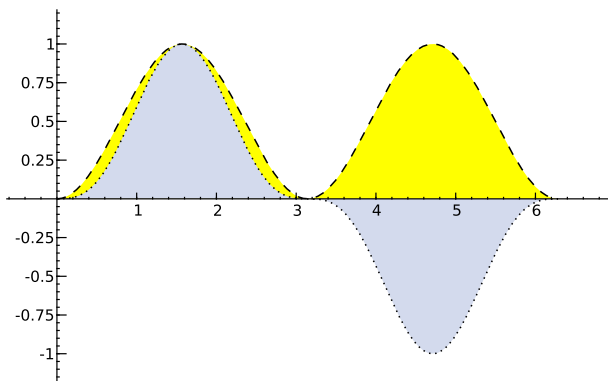
Pokud chceme zobrazit grafy více funkcí v jednom obrázku, máme několik možností. Můžeme například jako první parametr funkce `plot()` dát seznam vybraných funkcí (např. `plot([sin(x), sin(x**2)], -pi, pi)`), nastavení zobrazení jsou pak ale pro obě funkce společná a můžeme tak přijít o přehlednost.

```
sage: plot([sin(x), cos(x)], 0, 2*pi, fill=True, linestyle='--')
```



V případě, že potřebujeme pro každou funkci jiné nastavení, stačí příkaz `plot` použít dvakrát, mezi něj napsat jednoduše `+` a dosáhneme kýženého a většinou přehlednějšího výsledku (např. `plot(sin(x), -pi, pi, thickness = 2) + plot(sin(x**2), -pi, pi, color='red')`).

```
sage: r1 = plot(sin(x)^2, 0, 2*pi, color = (0,0,0), fill = True,
    fillcolor = "yellow", linestyle = '--')
sage: r2 = plot(sin(x)^3, 0, 2*pi, color = '#000000',
    fill = True, fillcolor = '#D3DAED', linestyle = ':')
sage: r1 + r2
```

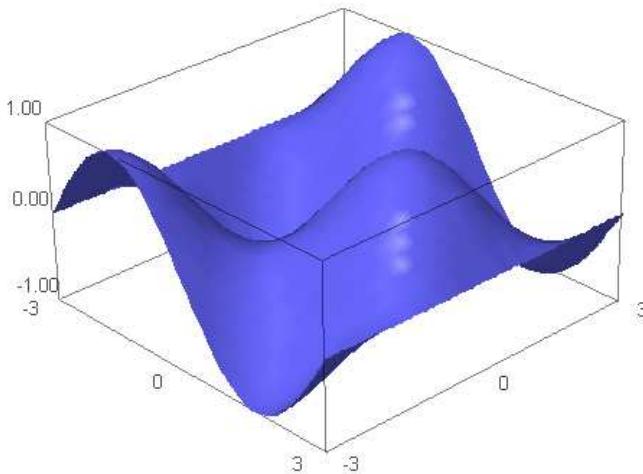


Grafy můžeme samozřejmě ukládat do externích souborů. Nejjednodušší možností je využít schopností internetového prohlížeče a po stisknutí pravého tlačítka myši na obrázku vybrat z kontextového menu možnost „Uložit obrázek jako...“ (přesný postup se bude samozřejmě lišit mezi jednotlivými prohlížeči). Obrázek potom máme možnost uložit v `.png` formátu. To s sebou ovšem přináší i určité nevýhody (např. pokud chceme využít obrázek v rámci textu sázeném v `LATEX`u) a proto je vhodnější využít funkci `save()`. Jako její první parametr pak můžeme dát předpis grafu, případně objekt, ve kterém máme graf uložen, a dále nastavit parametr `filename`, ve kterém zvolíme jméno souboru a zároveň i jeho formát. Celý příkaz pak může vypadat např. takto: `save(g1, filename="g1.eps")`.

5.2 Grafy funkcí dvou proměnných

Pro tvorbu grafů funkcí více proměnných slouží příkaz `plot3d()`. Prvním parametrem je opět předpis funkce, jejíž graf chceme vytvořit, jako další dva parametry je nutné určit rozmezí, ve kterém se budeme pohybovat po dvou vybraných osách, standardně x a y .

```
sage: y = var('y')
sage: plot3d(sin(x)*cos(y), (x,-pi,pi), (y,-pi,pi))
```



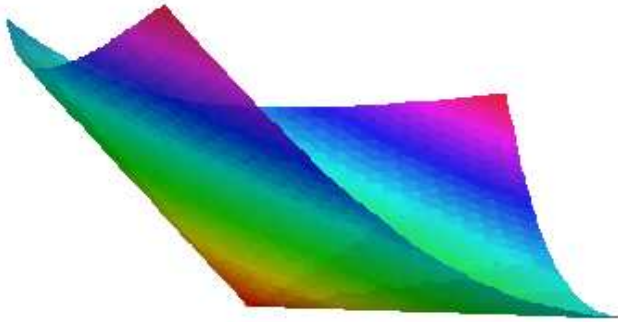
K vykreslení obrázku se využívá zobrazovací nástroj Jmol. Ten je vytvořen v programovacím jazyce Java a je tedy kvůli němu tomu třeba mít v počítači prostředí Java nainstalováno. Pokud ještě Javu v PC nainstalovanou uživatel nemá, může ji prostřednictvím stránek <http://www.java.com/en/download/index.jsp> stáhnout, poté nainstalovat a samozřejmě i v prohlížeči povolit její používání. S vzniklým grafem pak můžeme za pomoci myši libovolně pohybovat, obrázek přibližovat a přes kontextové menu (které nabízí dokonce možnost přepnutí do českého jazyka) různě nastavovat.

Spousta nastavení zobrazení se dá změnit už při tvorbě grafu pomocí parametrů funkce `plot3d()`:

- `color`: Nastavuje barvu plochy grafu, viz. stejný parametr u příkazu `plot`.
- `opacity`: Určuje průhlednost plochy grafu.
- `aspect_ratio`: Trojicí hodnot určuje poměr zobrazení jednotlivých dvojic os, např. `aspect_ratio=[1,1,1]`.
- `frame`: Přiřazením hodnoty `False` vypne zobrazení „boxu“ kolem grafu.
- `axes`: Nastavením na `True` zobrazí jednotlivé osy.
- `plot_points`: Určuje počet bodů, pro které se dopočítávají hodnoty funkce a z kterých se pak tvoří graf, např. `plot_points=[10,10]`.

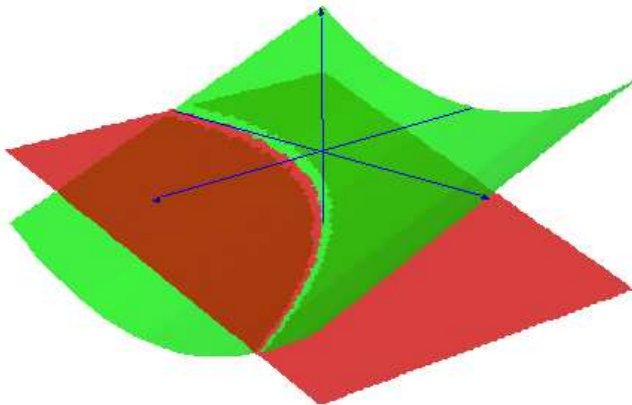
- `adaptive`: Při nezměněné výchozí hodnotě parametru `color` graf obarví vícebarevně podle „hloubky“.

```
sage: plot3d(sqrt(x**2 + y**2 + x*y), (x,-10,10), (y,-10,10),
            opacity = 0.87, adaptive = True, frame = False)
```



Zobrazení grafů více funkcí v jednom obrázku lze stejně jako u grafů funkcí jedné proměnné dosáhnout pomocí spojení dvou grafů znaménkem `+`. U příkazu `plot3d` však již nelze jako první parametr použít seznam více funkcí.

```
sage: f = x^2 + y^3 - 75*x*y
sage: fx = plot3d(diff(f, x), (x,-25,25), (y,-25,25),
            opacity = 0.7, color = 'red')
sage: fy = plot3d(diff(f, y), (x,-25,25), (y,-25,25),
            opacity = 0.7, color = 'green')
sage: show(fx + fy, axes = True, frame = False)
```



Ukládání obrázku grafu lze provést pomocí odkazu *Get Image* nacházejícího se vedle každého takového grafu. Ten nám z aktuálního pohledu na vykreslenou plochu vygeneruje JPEG soubor, který se ovšem moc nehodí pro použití v serióznějších dokumentech, neboť je poměrně dost komprimován. Příkaz `save()` nám tentokrát také nepomůže, ten totiž uloží graf ve formátu určeném pro práci s programem Jmol. Já jsem nakonec skončil u jednoduchého vytvoření snímku obrazovky z kterého jsem si požadovaný obrázek vyřízl v grafickém editoru. Tento postup ovšem evidentně také neposkytuje nejlepší výsledky.

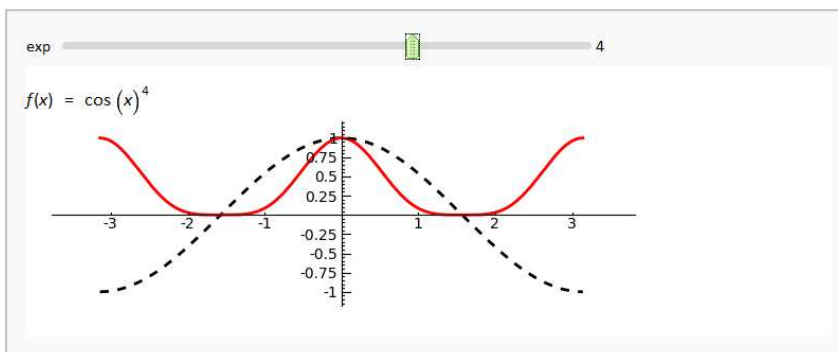
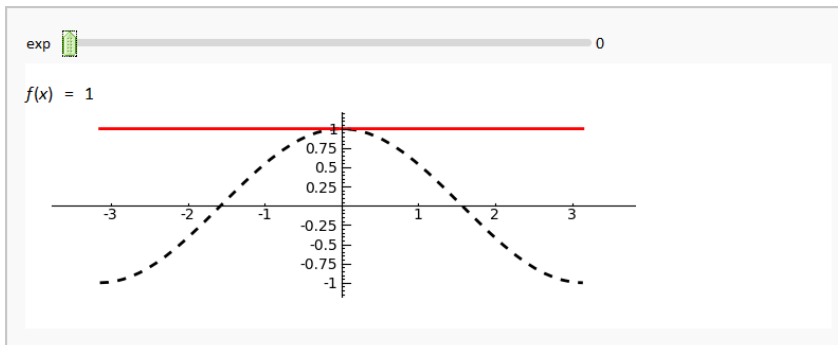
5.3 Interaktivní grafika

Sage nám umožňuje i tvorbu interaktivních obrázků vhodných především jako ukázky do výuky. Tato tvorba je již trochu obtížnější a je tak pro uživatele nejjednodušší (a autory dokonce doporučené) využití některé z mnoha šablon, které jsou k nalezení na stránce <http://wiki.sagemath.org/interact>. Po chvilce pak už každý zvládne přetvořit původní vzor do požadovaného tvaru.

Základem těchto interaktivních objektů je vždy nějaká proměnná, jejíž postupnou změnou na horní liště se mění i výsledek. Nemusí jít samozřejmě jen o grafy, možnosti interaktivity můžeme využít i pro běžné výpočty, ovšem použití právě v grafice je nejnázornější a pro výuku určitě nejzajímavější.

Nakonec zde máme jednoduchou ukázkou znázorňující vliv hodnoty parametru n na graf funkce $\cos(x)^n$:

```
sage: f = cos(x)
sage: g = plot(f, -pi, pi, thickness=2, color='black',
linestyle = '--')
sage: @interact
sage: def _(exp=(0..6)):
...     ge = plot(f^exp, -pi, pi, color='red', thickness=2)
...     html('$f(x)\;=\;\;%s\;%\latex(f^exp)')
...     show(g + ge, aspect_ratio=1)
```



Tato ukázka je v interaktivní podobě v vidění na stránce <http://www.sagenb.org/home/pub/255/>.

Kapitola 6

Lineární algebra

6.1 Vektory

Vektor se v systému Sage definuje příkazem `vector()`, jehož argumentem je seznam jeho souřadnic. S vektory pak můžeme provádět běžné operace jako je sčítání vektorů, násobení vektoru skalárem a vektorů mezi sebou a lineární kombinace vektorů.

```
sage: u = vector([2, -1, 0])
sage: v = vector([-1, 5, 3])
sage: u; v
(2, -1, 0)
(-1, 5, 3)
```

```
sage: u + v
(1, 4, 3)
```

```
sage: u*v
-7
```

```
sage: 3*u - 2*v
(8, -13, -6)
```

6.2 Matice

Pro zápis matic slouží příkaz `matrix()` obsahující jeden jediný argument a to seznam jednotlivých řádků matice zapsaných také jako seznamy. Vhodné matice pak mezi sebou můžeme násobit, sčítat, násobit vektory atp.

```
sage: A = matrix([[1, -6, 4, 0], [-4, 3, 8, 7]])
sage: A
[ 1 -6  4  0]
[-4  3  8  7]
```

```
sage: B = matrix([[2, -5], [3, 4], [-7, 1], [-2, 8]])
```

```
sage: B
[ 2 -5]
[ 3  4]
[-7  1]
[-2  8]
```

```
sage: A*B
[-44 -25]
[-69  96]
```

```
sage: B*A
[ 22 -27 -32 -35]
[-13  -6  44  28]
[-11  45 -20  7]
[-34  36  56  56]
```

```
sage: u = vector([1, 0, 4, 2])
sage: u*B
(-30, 15)
```

Transponovanou matici získáme pomocí funkce `transpose()`.

```
sage: transpose(A)
[ 1 -4]
[-6  3]
[ 4  8]
[ 0  7]
```

```
sage: transpose(A) + B
[ 3 -9]
[-3  7]
[-3  9]
[-2 15]
```

A nakonec hodnost matice nám zjistí funkce `rank()` a determinant spočte funkce `det()`.

```
sage: rank(A)
2
```

```
sage: rank(B*A)
2
```

```
sage: det(A*B)
-5949
```

6.3 Řešení rovnic a jejich soustav

Pro řešení rovnic slouží funkce `solve()`, kdy `solve(r, x)` řeší z rovnice `r` proměnnou `x`.

```
sage: solve(x^2 + 4*x - 2 == 0, x)
[x == -sqrt(6) - 2, x == sqrt(6) - 2]
```

Pokud chceme řešit soustavu více rovnic, dáme jako první parametr jejich seznam a jako druhý seznam řešených proměnných.

```
sage: x, y = var('x, y')
sage: solve([x - y == 2, x + y == 10], [x, y])
```

Sage stejně jako např. Maple nevrací vždy všechna řešení.

```
sage: solve(sin(x) == 1, x)
[x == pi/2]
```

Pokud chceme řešení vyjádřit numericky, je třeba nastavit volitelný parametr `solution_dict = True`, který upraví způsob uložení výsledků do seznamu, a poté na jednotlivé prvky seznamu výsledků aplikovat například pomocí for cyklu funkci `n()` pro numerickou aproximaci.

```
sage: p = solve(x^2 + 4*x == 2, x, solution_dict = True); p
[{x: -sqrt(6) - 2}, {x: sqrt(6) - 2}]
```

```
sage: for sol in p: print(n(sol[x], digits = 3))
-4.45
0.449
```

Závěr

Sage je přes své relativní mládí (přece jen je vyvíjen teprve od roku 2005) v mých očích velmi vyspělý a schopný nástroj. Nabízí uživateli možnost pracovat ve velmi přehledném a příjemném grafickém rozhraní, spoustu velmi dobře popsaných funkcí a to všechno zdarma. Nulová cena a otevřenost systému také stojí za tím, že kolem tohoto programu se již na internetu vytvořila rozsáhlá komunita, což samozřejmě posunuje vývoj neustále kupředu a i za těch pár měsíců, během kterých jsem s tímto softwarem pracoval, jsem si všiml velmi příjemných změn.

Samozřejmě že i Sage má hned několik záporů. Pro uživatele systému Windows je to především nutnost instalace virtualizačního softwaru třetí strany, ovšem tento „problém“ by měl být časem odstraněn díky vývoji nativní verze pro tento systém. Návoděda, především referenční příručka, je sice velmi obsáhlá, ovšem to s sebou nese i ten problém, že občas dá člověku poměrně zabrat najít to, co přesně hledá. A nakonec by se určitě našla i skupina lidí, která bude za zápor považovat neexistenci materiálů v našem rodném jazyce.

Já osobně jsem byl s prací se Sagem velmi spokojen, funguje rychle a spolehlivě, zmíněná virtualizace pod Windows také není velký problém (přesto se opravdu těším na vydání nativní verze), když člověk chvilku hledá, vždy najde mezi spoustou materiálů to, co hledá, a to vše je nabídnuto zdarma. Z mojí strany nemohu jinak než Sage doporučit minimálně k vyzkoušení např. přes stránky <http://www.sagenb.org/> a rozhodně s jeho využíváním sám nehodlám jen tak přestat.

Seznam použité literatury

- [1] *Computer algebra system - Wikipedia, the free encyclopedia.* (květen 2009).
http://en.wikipedia.org/wiki/Computer_algebra_system
- [2] *Virtuální stroj - Wikipedie, otevřená encyklopedie.* (květen 2009).
http://cs.wikipedia.org/wiki/Virtuální_stroj
- [3] *Welcome to Sage's Reference Manual! — Sage Reference Manual v3.4.* (květen 2009).
<http://www.sagemath.org/doc/reference/>
- [4] *Welcome to the Sage Tutorial! — Sage Tutorial v3.4.* (květen 2009).
<http://www.sagemath.org/doc/tutorial/>
- [5] *MyStartingPage - Sage Wiki.* (květen 2009).
<http://wiki.sagemath.org/>